

As per the Latest Syllabus of Anna University, Chennai
(Regulation - 2017)

Object Oriented Analysis and Design

With Latest
Solved
Anna university
Question Papers

For
B.E. / B.Tech
V Semester CSE
and
VI Semester IT
Branches

ARS PUBLICATIONS
CHENNAI.

P.Krishna Sankar
N.P.Shangaranarayanee

Table of Content

UNIT I.....
UNIFIED PROCESS AND USE CASE DIAGRAMS.....
1.1 Introduction to OOAD with OO Basics.....
1.1.1 Objects and Classes.....
1.1.2 Analysis.....
1.1.3 Design.....
1.1.4 Object-Oriented Analysis.....
1.1.5 Object-Oriented Design.....
1.1.6 Object-Oriented Analysis and Design.....
1.2 Unified Process.....
1.2.1 Most Important UP Idea: Iterative Development.....
1.2.2 UP Phases and Schedule-Oriented Terms.....
1.2.3 UP Disciplines (was Workflows).....
1.2.4 Process Customization and the Development Case.....
1.2.5 Agile UP.....
1.2.6 Benefits of Iterative Development.....
1.3 UML Diagrams.....
1.3.1 Introduction.....
1.3.2 UML Building Blocks.....
1.3.3 Types of UML Diagram.....
1.3.4 Role of UML in OO design.....
1.3.5 Goals of UML.....
1.4 Use Case.....
1.4.1 Introduction.....
1.4.2 Purpose.....
1.4.3 Use Case Notations.....
1.4.4 When to Use Use Cases.....
1.4.5 Examples.....

1.5	Case Study - The Next Gen Pos System, Inception.....
1.5.1	Architectural Layers and Case Study Emphasis.....
1.5.2	User-Level Goals.....
1.6	Use Case Modelling.....
1.6.1	Introduction.....
1.6.2	Goals and Stories.....
1.6.3	Use Cases and Adding Value.....
1.6.4	Use Cases and Functional Requirements
1.6.5	Use Case Types and Formats.....
1.6.6	Fully Dressed Example: Process Sale.....
1.6.7	Explaining the Sections.....
1.6.8	Goals and Scope of a Use Case.....
1.6.9	Finding Primary Actors, Goals, and Use Cases.....
1.6.10	Congratulations: Use Cases Have Been Written, and Are Imperfect.....
1.6.11	Write Use Cases in an Essential ILL-Free Style New and Improved! The Case for Fingerprinting.....
1.6.12	Actors.....
1.6.13	Use Case Diagrams.....
1.6.14	Requirements in Context and Low-Level Feature Lists.....
1.6.15	Use Cases Are Not Object-Oriented.....
1.6.16	Use Cases Within the UP.....
1.6.17	Case Study: Use Cases in the NextGen Inception Phase.....
1.6.18	UP Artifacts and Process Context.....
1.7	Relating Use Cases.....
1.7.1	Introduction.....
1.7.2	The include Relationship.....
1.7.3	The extend Relationship.....
1.7.4	The generalize Relationship.....
1.8	When to Use Use-Cases.....
	Review Questions.....

UNIT 2.....	
STATIC UML DIAGRAMS.....	
2.1	Class Diagram.....
2.1.1	Introduction.....
2.1.2	Purpose.....
2.1.3	Class Diagram Notations.....
2.1.4	When to use: Class Diagram.....
2.1.5	Examples.....
2.2	Elaboration.....
2.2.1	Introduction.....
2.2.2	What Is Architecturally Significant in Elaboration?
2.2.3	Planning the Next Iteration.....
2.2.4	Iteration 1 Requirements and Emphasis: Fundamental OOA/D Skills.....
2.2.5	Elaboration Artifacts.....
2.3	Domain Model.....
2.3.1	Introduction.....
2.3.2	Key Idea: Domain Model—A Visual Dictionary of Abstractions.....
2.3.3	Domain Models Are not Models of Software Components.....
2.3.4	Example: The NextGen POS Domain Model.....
2.4	Finding conceptual classes and description classes.....
2.4.1	Conceptual Class Identification.....
2.4.2	Candidate Conceptual Classes for the Sales Domain.....
2.4.3	Domain Modeling Guidelines.....
2.4.4	Resolving Similar Conceptual Classes—Register vs. "POST"
2.4.5	Modeling the Unreal World.....
2.4.6	Specification or Description Conceptual Classes.....
2.4.7	UML Notation, Models, and Methods: Multiple Perspectives.....
2.4.8	Lowering the Representational Gap.....
2.5	Associations.....
2.5.1	What are association relationships?
2.5.2	UML notation for association.....

2.5.3	Using verbs to identify associations.....
2.5.4	Role names.....
2.5.5	Use interfaces or the server side of associations.....
2.5.6	Association constraints.....
2.5.7	Association classes.....
2.5.8	Qualifications.....
2.5.9	N-ary associations.....
2.5.10	Simplify N-ary associations by introducing a mediator.....
2.5.11	Avoid spaghetti communication networks.....
2.6	Attributes.....
2.6.1	Introduction.....
2.6.2	Attributes.....
2.6.3	UML Attribute Notation.....
2.6.4	Valid Attribute Types.....
2.6.5	Non-primitive Data Type Classes.....
2.6.6	Design Creep: No Attributes as Foreign Keys.....
2.6.7	Modeling Attribute Quantities and Units.....
2.6.8	Attributes in the NextGen Domain Model.....
2.6.9	Multiplicity from SalesLineItem to Item.....
2.6.10	Domain Model Conclusion.....
2.7	Domain model refinement.....
2.7.1	Introduction.....
2.7.2	Association Classes.....
2.7.3	Aggregation and Composition.....
2.7.4	Time Intervals and Product Prices—Fixing an Iteration 1 "Error"
2.7.5	Association Role Names
2.7.6	Derived Elements.....
2.7.7	Association Role Names.....
2.7.8	Qualified Associations.....
2.7.9	Reflexive Associations.....
2.7.10	Ordered Elements.....

2.7.11	Using Packages to Organize the Domain Model.....
2.7.12	POS Domain Model Packages.....
2.8	Finding conceptual class Hierarchies.....
2.8.1	New Concepts for the Domain Model.....
2.8.2	Generalization.....
2.8.3	Defining Conceptual Superclasses and Subclasses.....
2.8.4	When to Define a Conceptual Subclass.....
2.8.5	When to Define a Conceptual Superclass.....
2.8.6	NextGen POS Conceptual Class Hierarchies.....
2.8.7	Abstract Conceptual Classes.....
2.8.8	Modeling Changing States.....
2.8.9	Class Hierarchies and Inheritance in Software.....
2.9	Aggregation and Composition.....
2.9.1	Aggregation in the UML.....
2.9.2	Composite Aggregation—Filled Diamond.....
2.9.3	Shared Aggregation—Hollow Diamond.....
2.9.4	Composition.....
2.9.5	A Benefit of Showing Composition.....
2.9.6	Composition in the NextGen Domain Model.....
2.10	Relationship between sequence diagrams and use cases
2.10.1	Inter-System SSDs.....
2.10.2	SSDs and Use Cases.....
2.10.3	System Events and the System Boundary.....
2.10.4	Naming System Events and Operations.....
2.10.5	Showing Use Case Text.....
2.10.6	SSDs and the Glossary.....
2.11	When to use Class Diagrams.....
	Review Questions.....

UNIT III.....

DYNAMIC AND IMPLEMENTATION UML DIAGRAMS.....

3.1	Dynamic and Implementation Diagrams.....
3.2	Interaction diagrams.....
3.2.1	Sequence Diagrams.....
3.2.2	Collaboration Diagrams.....
3.2.3	Comparing Sequence and Collaboration Diagrams.....
3.2.4	CRC Cards.....
3.3	When to use Communication Diagrams.....
3.4	State machine diagram and Modelling.....
3.4.1	Introduction.....
3.4.2	Purpose.....
3.4.3	State Diagram Notations.....
3.4.4	When to use: State Diagram.....
3.4.5	Examples.....
3.5	Activity diagram.....
3.5.1	Introduction.....
3.5.2	Purpose.....
3.5.3	Activity diagrams Notation.....
3.5.4	When to use: Activity Diagrams.....
3.6	Package.....
3.6.1	Introduction.....
3.6.2	Collaborations.....
3.6.3	When to Use Package Diagrams and Collaborations.....
3.7	Component and Deployment Diagrams.....
3.7.1	Deployment Diagrams.....
3.7.2	Component Diagrams.....
3.7.3	Combining Component and Deployment Diagrams.....
3.7.4	When to Use Physical Diagrams.....
	Review Questions.....

UNIT IV

DESIGN PATTERNS

4.1	GRASP: Designing objects with responsibilities.....
4.1.1	Introduction.....
4.1.2	GRASP as a Methodical Approach to Learning Basic Object Design.....
4.1.3	Responsibilities and Methods.....
4.1.4	Patterns.....
4.1.5	GRASP: Patterns of General Principles in Assigning Responsibilities.....
4.1.6	The UML Class Diagram Notation.....
4.2	Creator.....
4.2.1	Solution.....
4.2.2	Problem.....
4.2.3	Example.....
4.2.4	Discussion.....
4.2.5	Contraindications.....
4.2.6	Benefits.....
4.2.7	Related Patterns or Principles.....
4.3	Information expert.....
4.3.1	Solution.....
4.3.2	Problem.....
4.3.3	Example.....
4.3.4	Discussion.....
4.3.5	Contraindications.....
4.3.6	Benefits.....
4.3.7	Related Patterns or Principles.....
4.4	Low Coupling.....
4.4.1	Solution.....
4.4.2	Problem.....
4.4.3	Example.....
4.4.4	Discussion.....
4.4.5	Contraindications.....
4.4.6	Benefits.....
4.4.7	Background.....

4.4.8	Related Pattern
4.5	High Cohesion.....
4.5.1	Solution.....
4.5.2	Problem.....
4.5.3	Example.....
4.5.4	Discussion.....
4.5.5	Contraindications.....
4.5.6	Benefits.....
4.6	Controller.....
4.6.1	Solution.....
4.6.2	Problem.....
4.6.3	Example.....
4.6.4	Discussion.....
4.6.5	Benefits.....
4.7	Design Patterns.....
4.7.1	What is a Design Pattern?
4.7.2	Design Patterns in Smalltalk MVC.....
4.7.3	Describing Design Patterns.....
4.7.4	How to Select a Design Pattern?
4.8	Creational.....
4.9	Factory method
4.9.1	Intent.....
4.9.2	Motivation.....
4.9.3	Applicability.....
4.9.4	Structure.....
4.9.5	Participants.....
4.9.6	Collaborations.....
4.9.7	Consequences.....
4.9.8	Implementation.....
4.10	Structural.....
4.11	Bridge.....

4.11.1	Intent.....
4.11.2	Motivation.....
4.11.3	Applicability.....
4.11.4	Structure.....
4.11.5	Participants.....
4.11.6	Collaborations.....
4.11.7	Consequences.....
4.11.8	Implementation.....
4.11.9	Known Uses.....
4.11.10	Related Patterns.....
4.12	Adapter.....
4.12.1	Intent.....
4.12.2	Motivation.....
4.12.3	Applicability.....
4.12.4	Structure.....
4.12.5	Participants.....
4.12.6	Collaborations.....
4.12.7	Consequences.....
4.12.8	Implementation.....
4.12.9	Known Uses.....
4.12.10	Related Patterns.....
4.13	Behavioural.....
4.14	Strategy.....
4.14.1	Intent.....
4.14.2	Motivation.....
4.14.3	Applicability.....
4.14.4	Structure.....
4.14.5	Participants.....
4.14.6	Collaborations.....
4.14.7	Consequences.....
4.14.8	Implementation.....

4.14.9	Known Uses.....
4.14.10	Related Patterns.....
4.15	Observer.....
4.15.1	Intent.....
4.15.2	Motivation.....
4.15.3	Applicability.....
4.15.4	Structure.....
4.15.5	Participants.....
4.15.6	Collaborations.....
4.15.7	Consequences.....
4.15.8	Implementation.....
4.15.9	Known Uses.....
4.15.10	Related Patterns.....
4.16	Applying GoF design patterns.....
4.16.1	The Gang-of-Four Design Patterns.....
4.16.2	Adapter(GOF)
4.16.3	Polymorphism, Indirection, and Protected Variations (GRASP).....
4.16.4	Naming Convention: Embed Pattern Name in Type Name?
4.16.5	"Analysis" Discoveries During Design: Domain Model.....
4.16.6	Factory (GoF)
4.16.7	Singleton (GoF)
4.16.8	Strategy (GoF)
4.16.9	Composite (GoF) and Other Design Principles.....
4.17	Mapping design to code.....
4.17.1	Programming and the Development Process.....
4.17.2	Mapping design to code.....
4.17.3	Defining the Sale--makeLineItem Method.....
4.17.4	Order of Implementation.....
4.17.5	Test Driven or Test-First Programming.....
	Review Questions.....

UNIT V
TESTING.....
5.1 Object Oriented Methodologies.....
5.1.1 What is Object Oriented Methodology?
5.1.2 Objectives of Object Oriented Methodologies.....
5.1.3 Benefits of Object Oriented Methodologies.....
5.2 Software Quality Assurance.....
5.2.1 Definition.....
5.2.2 Objectives.....
5.3 Impact of object orientation on Testing.....
5.3.1 Object-oriented Testing Methods.....
5.3.2 Challenges in Testing Object-oriented Programs.....
5.4 Develop Test Cases and Test Plans.....
5.4.1 What is a Test Plan?
5.4.2 Test Plan Identifiers.....
5.4.3 Test Planning Activities.....
5.4.4 Developing the test cases.....
5.4.5 Step by Step guide to Test Case Development.....
5.4.6 Activities in the Test Case Development Phase.....
Review Questions.....
Two mark Questions and Answers.....
Model Question Papers.....